IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sahana Reddy P
12-07-2022

Outline

- Executive Summary
- Introduction
- Methodology
- Data insights
- Results
- Conclusion
- Appendix

Executive Summary

Below are the data science methodologies that were followed to determine if the first stage of Falcon9 rocket launch will land successfully :

- Data Collection
- Data Cleansing
- Exploratory Data Analysis Using SQL
- Exploratory Data Analysis Using Pandas and Matplotlib
- Building interactive Dashboard using Folium , Plotly and Dash Framework
- Predictive Analysis Using Classification Models
- Decide the best Model based on accuracy
- Conclusion

Introduction

Current state

- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars while other providers cost upward of 165 million dollars each.
- The reason for most of the savings being SpaceX can reuse the first stage

Desired state

- The task of this project is to determine if the first stage of the SpaceX Falcon9 rocket will land successfully based on which the cost of a launch can be determined.

Methodology

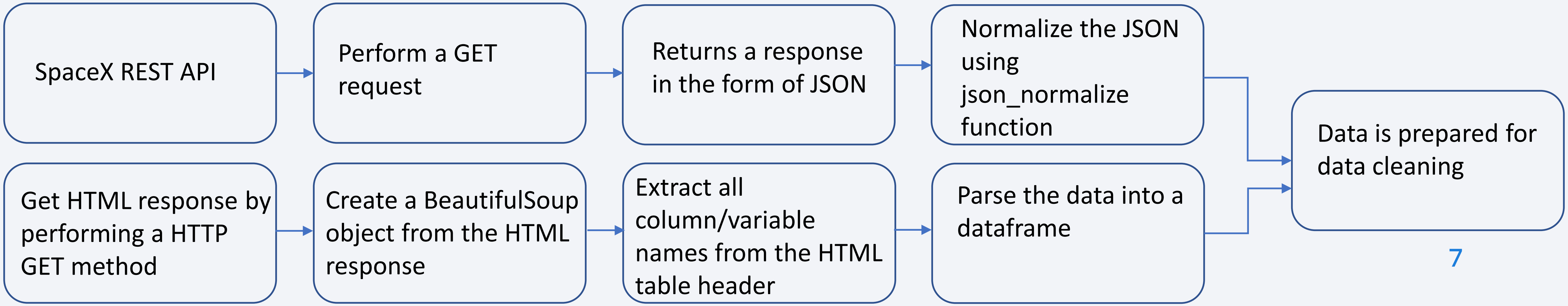
Methodology

Executive Summary

- Data collection methodology :
    - The Data was collected using SpaceX API
    - Web scraping from Wikipedia using BeautifulSoup package
- Perform data wrangling
    - Data cleaning by identifying the Null values and irrelevant columns.
    - Created a landing outcome label (1 - First stage landed successfully , 0 – First stage did not land successfully)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Standardized the data , Built SVM , Decision Tree and Logistic Regression model and found the best classifier based on the accuracy metrics.

Data Collection

- The SpaceX launch data was collected using
  - SpaceX REST API
  - The API will provide the data about launches, rocket used, payload delivered , launch specifications , landing specifications and landing outcome
  - The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/
  - Another method of data collection is Web scraping from Wikipedia using BeautifulSoup library in Python



Data Collection – SpaceX API

- The SpaceX launch data was collected using SpaceX REST API
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/
- GitHub link for Data Collection using API

1. Perform a GET request to get a response from REST API

```
In [31]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [32]: response = requests.get(spacex_url)
```

2. Normalize the JSON using json_normalize function

```
In [36]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
In [12]: # Get the head of the dataframe
         data.head()
```

Data Collection – SpaceX API

3. Apply custom functions to clean the data

```
In [40]: # Call getBoosterVersion
getBoosterVersion(data)
```

the list has now been update

```
In [41]: BoosterVersion[0:5]

Out[41]: ['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

```
In [42]: # Call getLaunchSite
getLaunchSite(data)

In [43]: # Call getPayloadData
getPayloadData(data)

In [44]: # Call getCoreData
getCoreData(data)
```

4. Assign the lists to dictionary **launch_dict**

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

Data Collection - Scraping

- Web scrap the data from wikipedia
- GitHub link for Data collection using Web scraping

1. Getting Response from HTML

In [48]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

2. Create a BeautifulSoup object

In [49]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content,"html.parser")
```

3. Finding all the tables

In [51]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
html_tables = soup.find_all("table")
# Assign the result to a list called `html_tables`
```

4. Getting column names

In [53]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
ths =  first_launch_table.find_all('th')
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for th in ths:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)

# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
```

Data Collection - Scraping

- Web scrap the data from wikipedia
- GitHub link for Data collection using Web scraping

5. Defining a dictionary

In [99]:
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to dictionary keys (Refer to **TASK 3: Create a data frame by parsing the launch HTML tables** heading in the notebook) and then converting to dictionary

In [101…
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

In [105…
```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

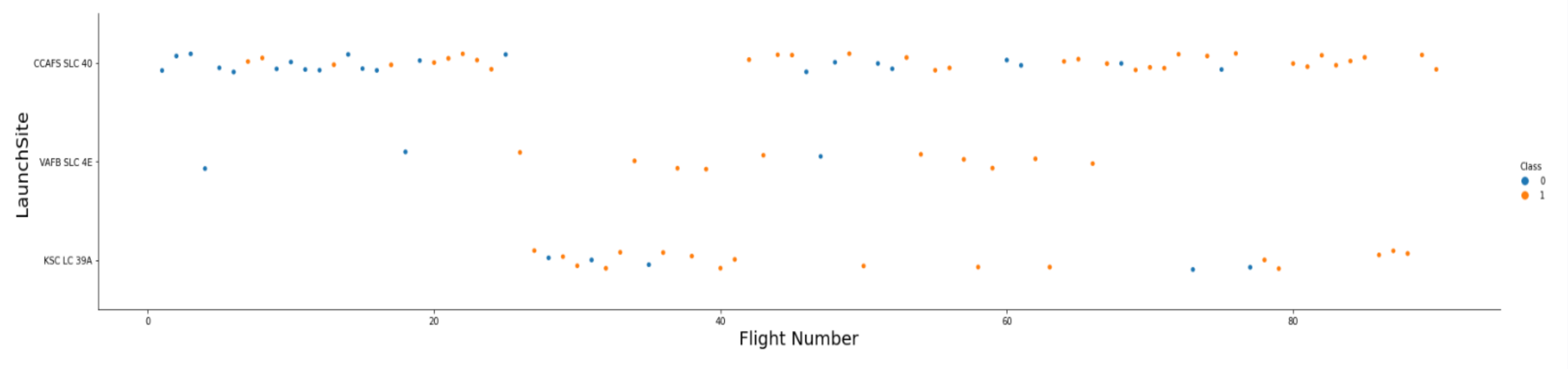
- Data cleaning by identifying the Null values and irrelevant columns.
- Created a landing outcome label (1 - First stage landed successfully , 0 – First stage did not land successfully)

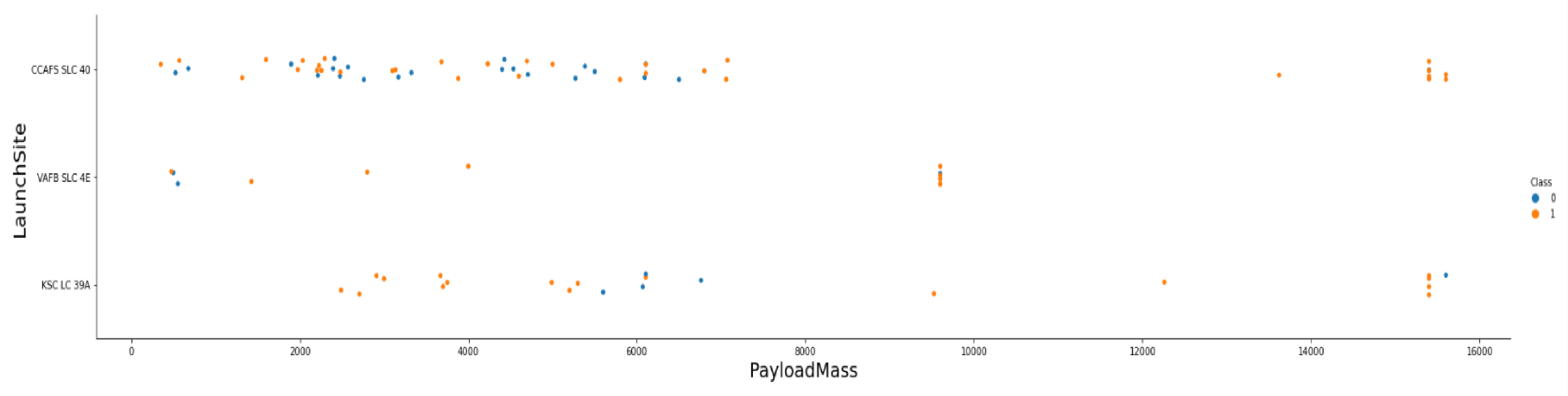
1. Check for Missing values in each field →
2. Calculate the number of launches on each site →
3. Calculate the occurrence of each orbit →
4. Calculate the number and occurrence of mission outcome per orbit type →
5. Create a landing outcome label from Outcome column

- https://github.com/Sahana-1995/Applied-Capstone-project/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

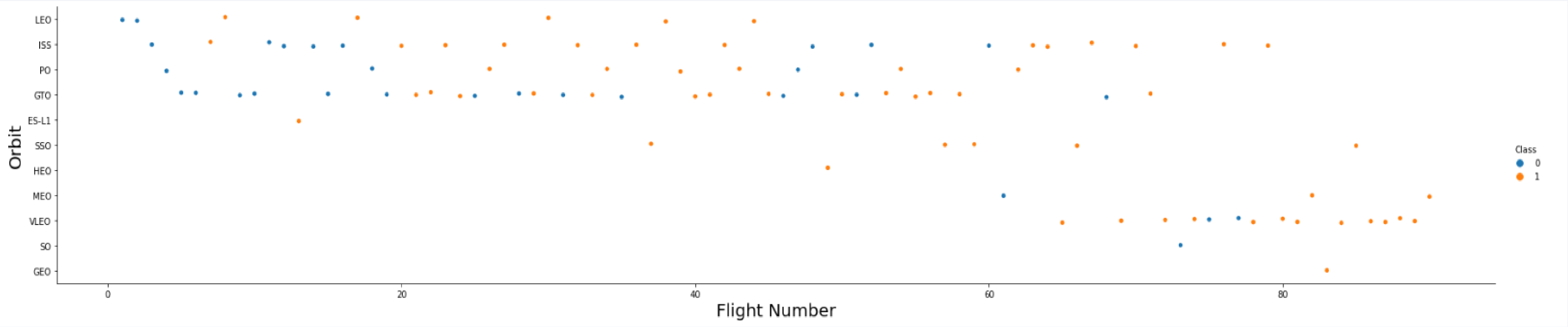
- To visualize the occurrence of the binary outcome with respect to the 2 independent variables

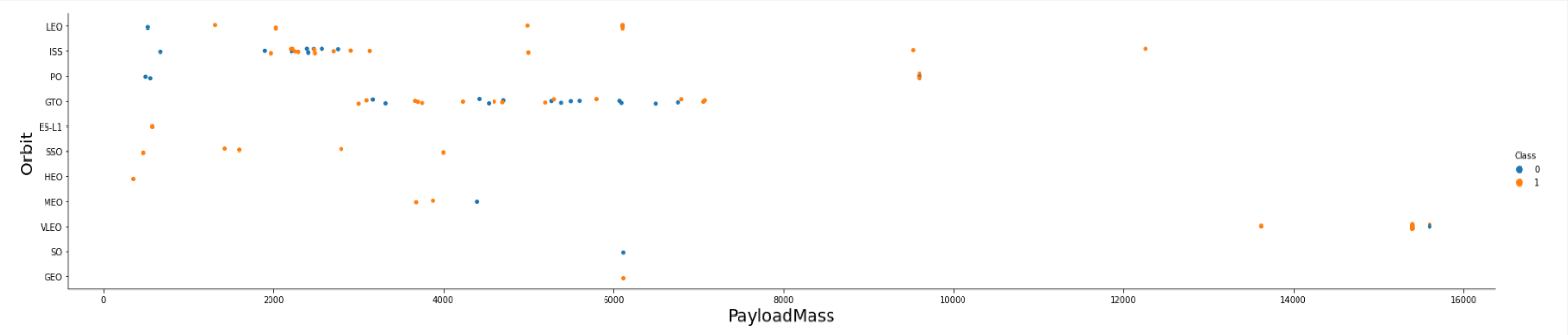




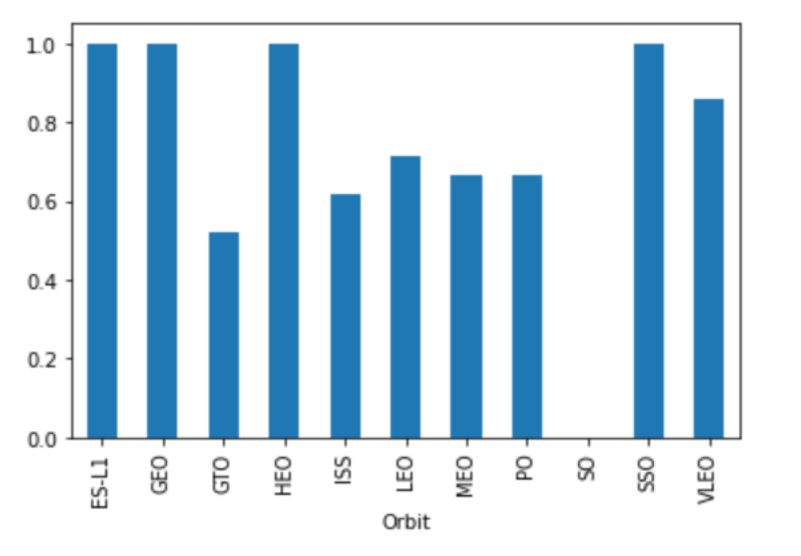
EDA with Data Visualization

- To visualize the occurrence of the binary outcome with respect to the 2 independent variables

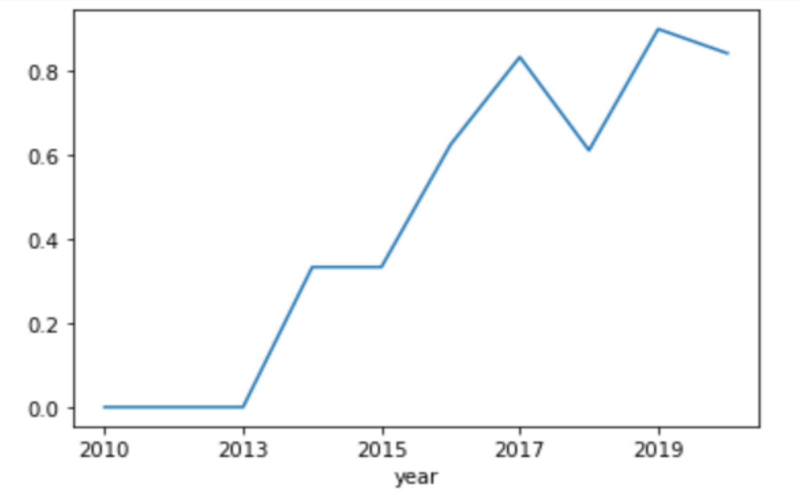




EDA with Data Visualization



- To visualize the average success rate for each orbit



- To visualize the average launch success trend

https://github.com/Sahana-1995/Applied-Capstone-project/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission

  select distinct(Launch_Site) from SPACEXTBL

- Display 5 records where launch sites begin with the string 'CCA'

  select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5

- Display the total payload mass carried by boosters launched by NASA (CRS)

  select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = "NASA (CRS)"

- Display average payload mass carried by booster version F9 v1.1

  select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like "F9 v1.1%"

- List the date when the first successful landing outcome in ground pad was achieved.

  select min(Date) from SPACEXTBL where "Landing _Outcome" = "Success (ground pad)"

EDA with SQL

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

select distinct Booster_Version from SPACEXTBL where "Landing _Outcome" = "Success (drone ship)"

and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_< 6000

- List the total number of successful and failure mission outcomes

select Mission_Outcome, count(Mission_Outcome) from SPACEXTBL group by Mission_Outcome

- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

select distinct Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

select substr(Date, 4, 2) as month,Booster_Version,Launch_Site,count("Landing _Outcome") from SPACEXTBL where substr(Date,7,4)='2015' and "Landing _Outcome" = "Failure (drone ship)" group by 1,2,3

EDA with SQL

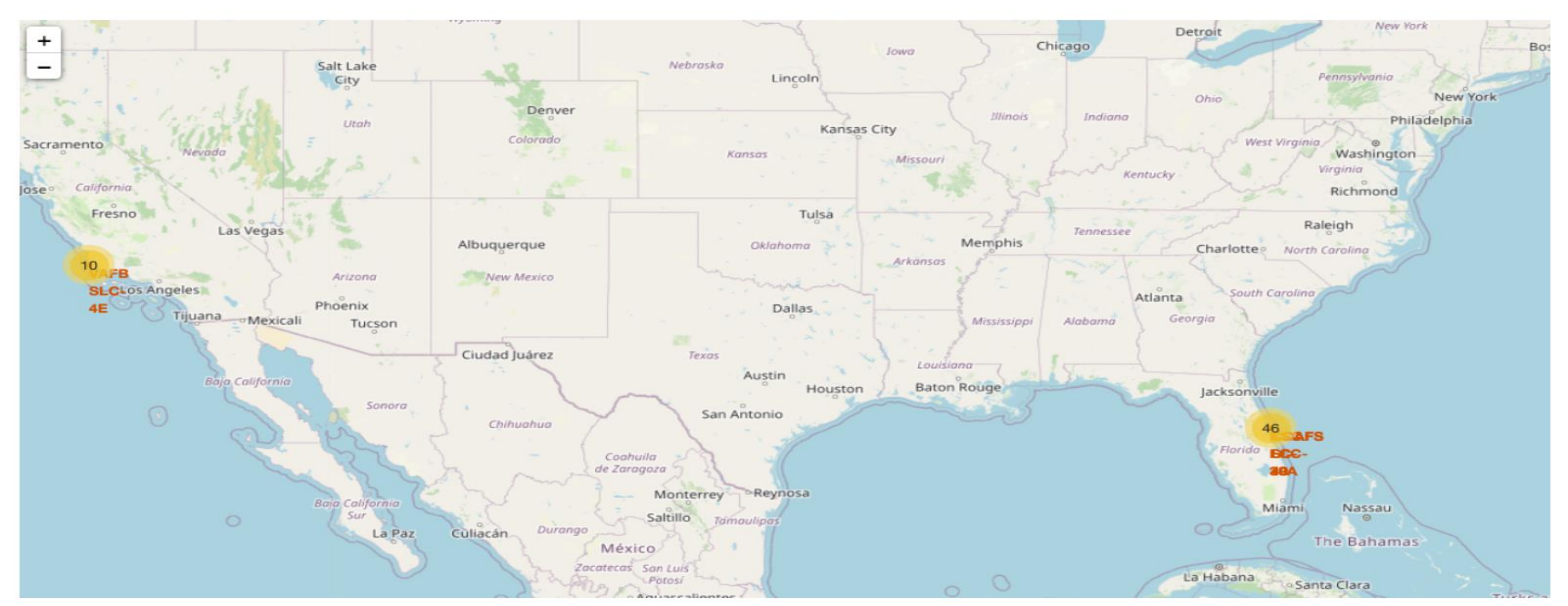
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

select "Landing _Outcome" , count("Landing _Outcome") from SPACEXTBL where Date between '04-06-2010' and '20-03-2017'

and "Landing _Outcome" like '%Success%’ group by "Landing _Outcome“ order by count("Landing _Outcome") desc

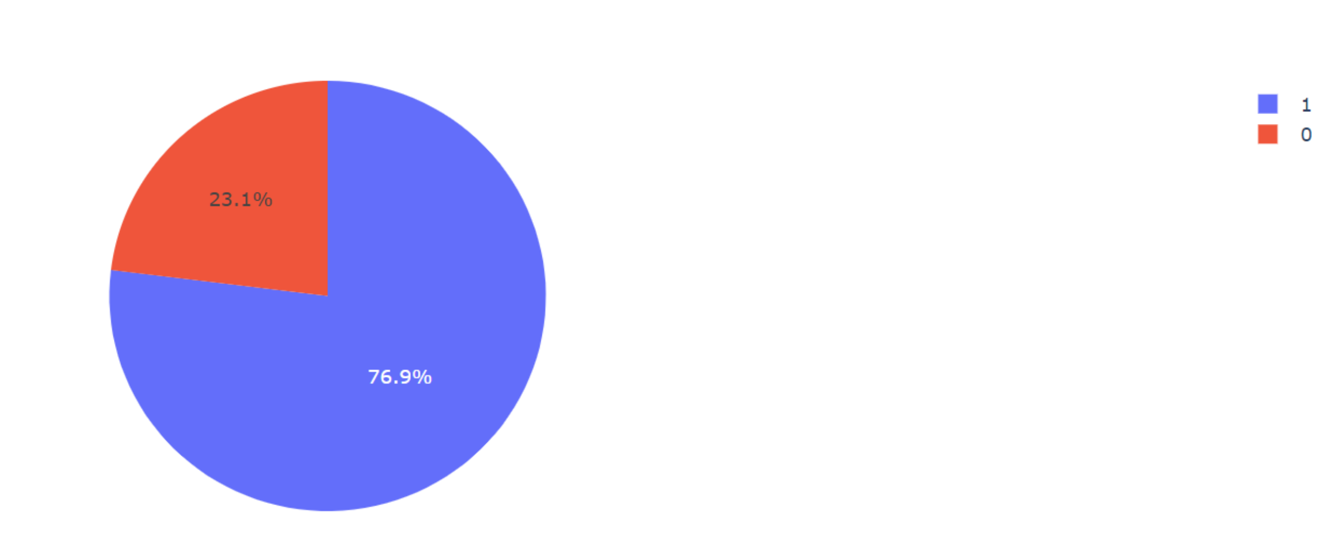
https://github.com/Sahana-1995/Applied-Capstone-project/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium



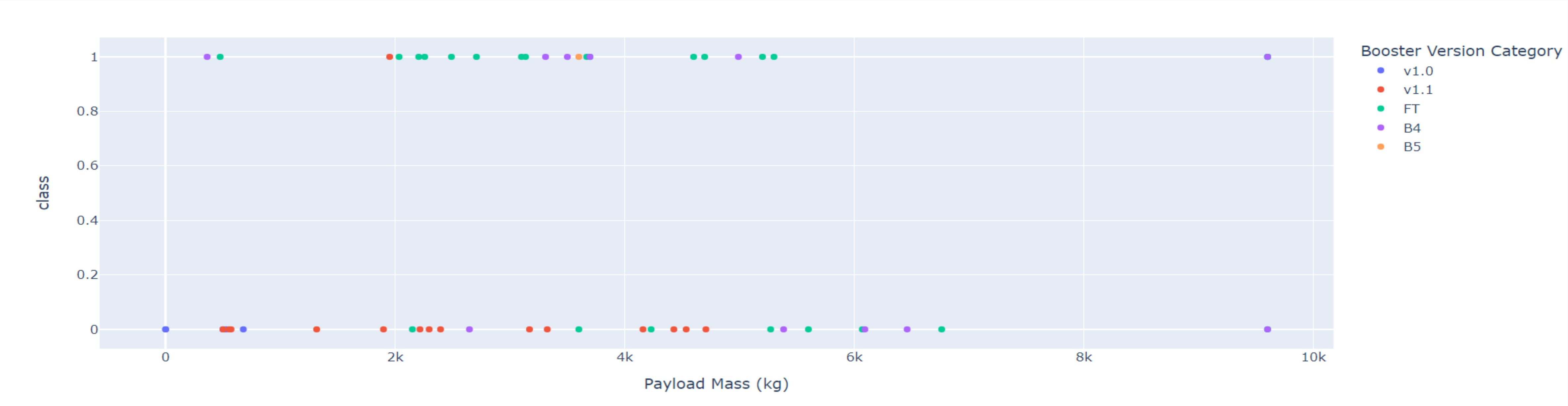
- Map markers have been added to the map in order to find the most successful launch site and their proximities to railways , highways , coastline and cities
- https://github.com/Sahana-1995/Applied-Capstone-project/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash



- KSC LC-39A had the most successful launches among all the other sites
- KSC LC-39A achieved 76.9% of success rate while getting a failure rate of 23.1%

Build a Dashboard with Plotly Dash

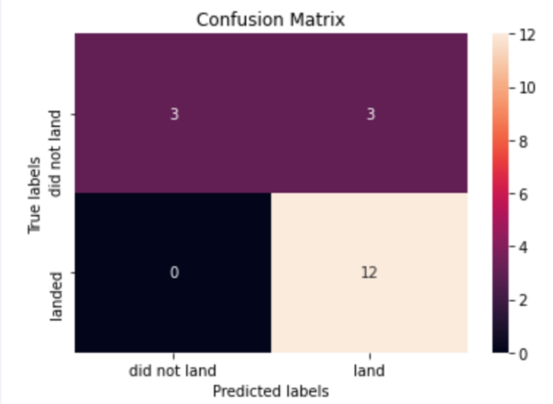


- Success rates for low weighted payloads is higher than the heavy weighted payloads

https://github.com/Sahana-1995/Applied-Capstone-project/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- Built Logistic Regression , KNN , Decision Tree and SVM Models
- LR, KNN and SVM achieved the highest accuracy of 83.33% and below is the confusion matrix plot for all the models
- https://github.com/Sahana-1995/Applied-Capstone-project/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Define variable **Y** from the column **Class** → Standardize the data and assign it to variable X → Split the data in X and Y into training and test data → Create a LR , SVM , Decision Tree and KNN object → Create a GridSearchCV object → Fit the GridSearchCV on training data for LR, SVM , DT and KNN → Find the best hyper parameters using GridSearchCV → Calculate the accuracy of the models using test data → Compare the accuracy to decide the best model

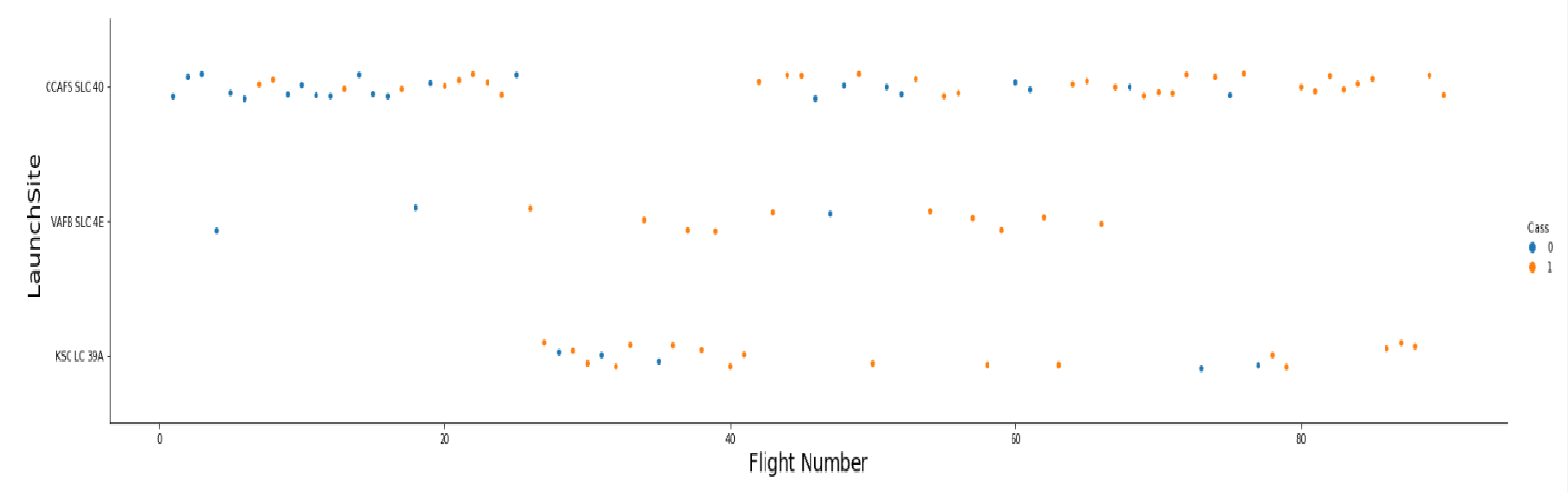
Results

- ES-L1, GEO, HEO and SSO orbits have the highest success rates
- The SpaceX launch success rate has been almost consistently increasing since 2013
- KSC LC-39A had the most successful launches among all the other sites
- Success rates for low weighted payloads is higher than the heavy weighted payloads
- The LR, KNN and SVM achieved the highest accuracy of 83.33%

Section 2

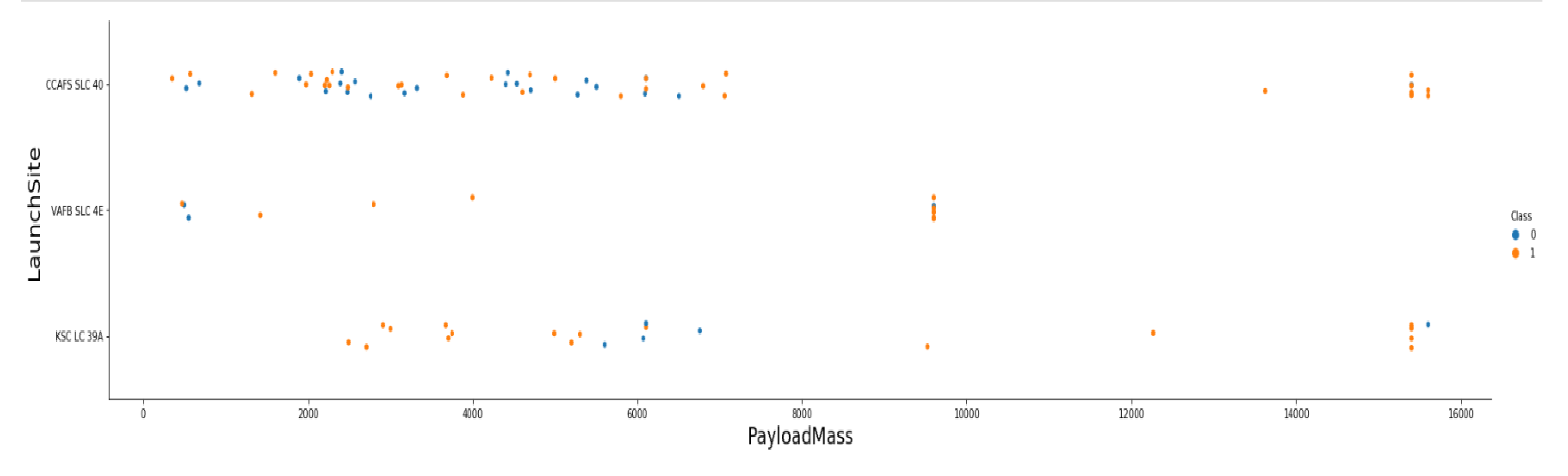
Insights drawn from EDA

Flight Number vs. Launch Site



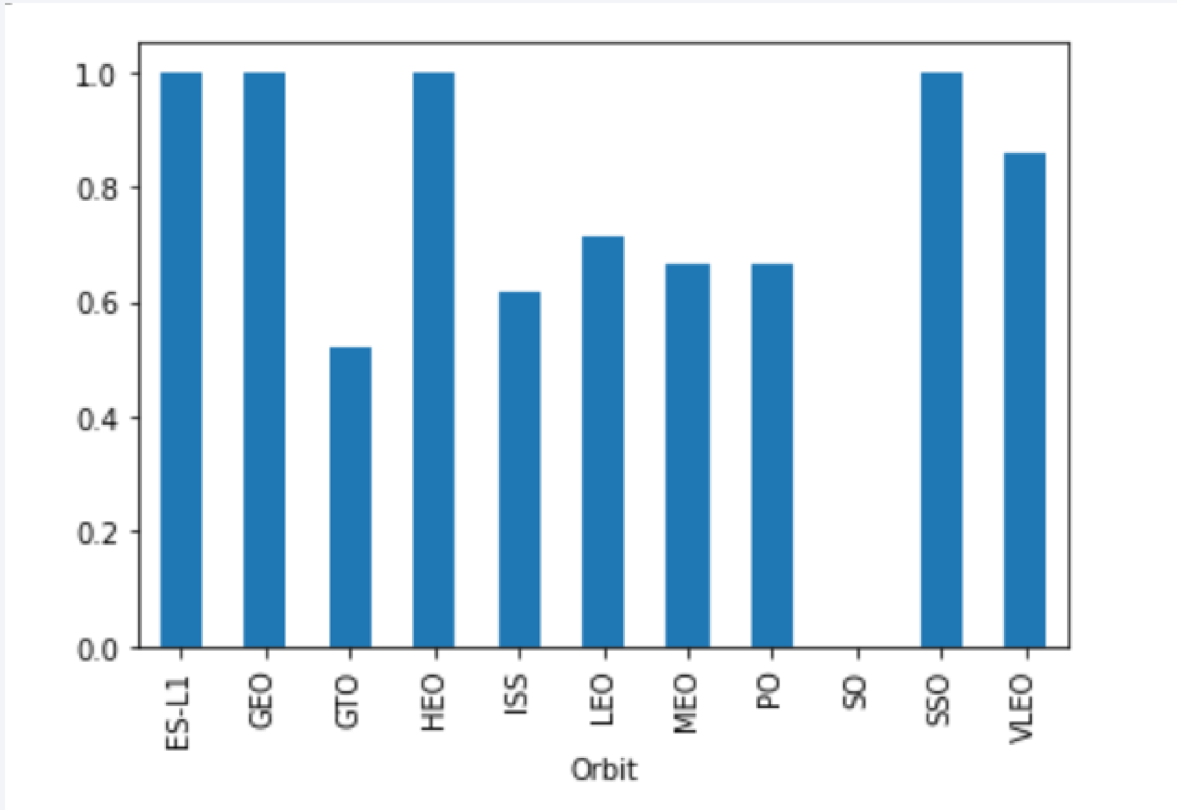
- Launches from the site CCAFS SLC 40 are significantly higher than the launches from other sites.

Payload vs. Launch Site



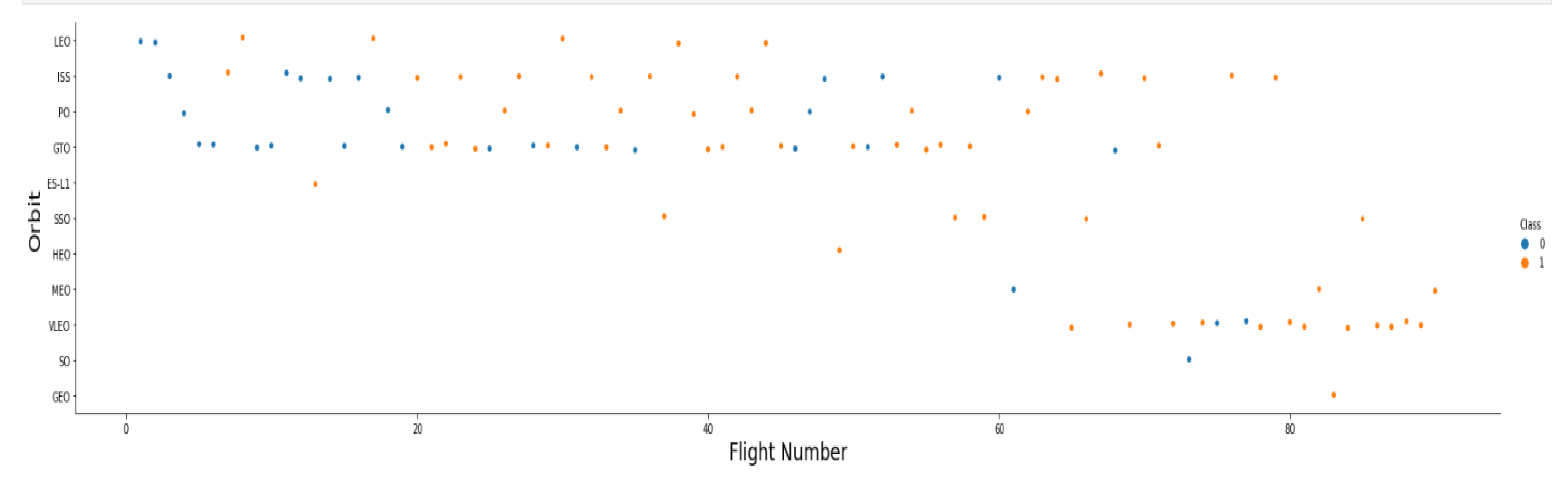
- There are no rockets launched in VAFB-SLC launch site with heavy payload mass(greater than 10000)
- CCAFS SLC 40 launch site has majority of the launches with lower payload mass

Success Rate vs. Orbit Type



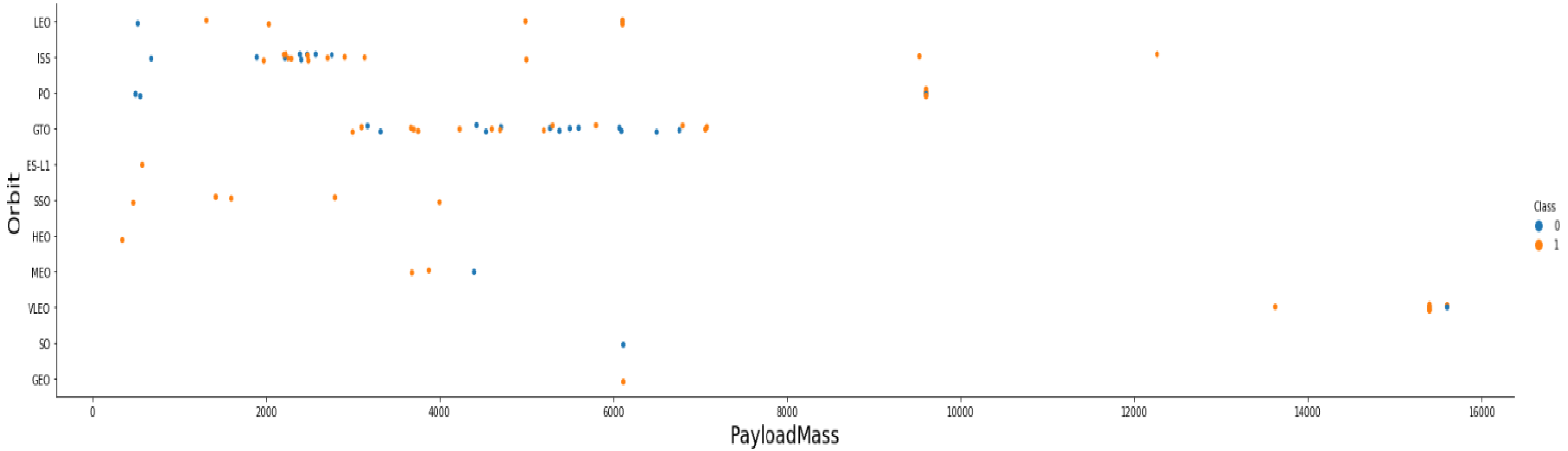
- ES-L1, GEO, HEO and SSO orbits have the highest success rates

Flight Number vs. Orbit Type



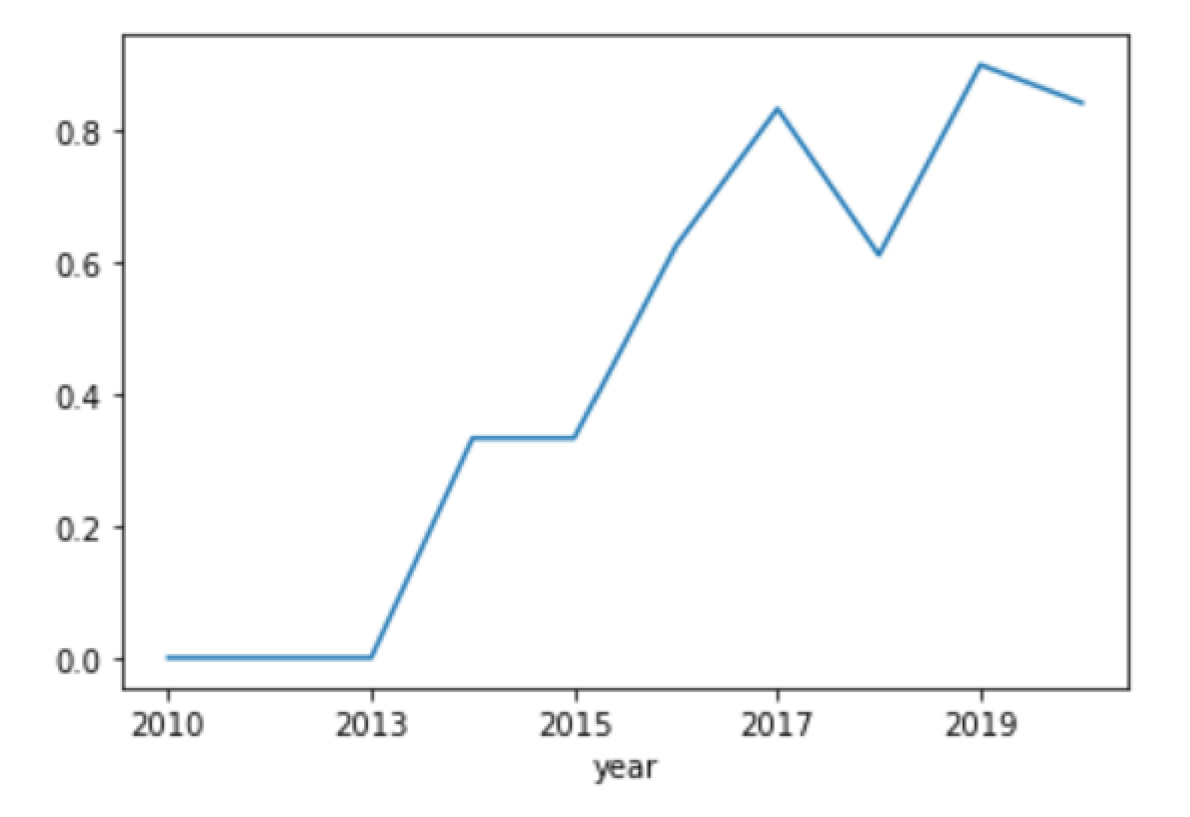
- In the LEO orbit the Success appears to be related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing rate are more for Polar , LEO and ISS. However, for GTO we cannot distinguish between the positive landing and negative landing rate

Launch Success Yearly Trend



- Success rate has increased significantly since 2013 except for a drop in 2018

All Launch Site Names

- The unique launch sites are :
  - CCAFS SLC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS LC-40

In [12]:
```
%%sql
select distinct(Launch_Site) from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Out[12]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

In [17]:

```
%%sql
select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

Out[17]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
In [18]: %%sql
select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = "NASA (CRS)"

 * sqlite:///my_data1.db
Done.
```

Out[18]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
In [21]: %%sql
         select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like "F9 v1.1%"

 * sqlite:///my_data1.db
Done.
```

Out[21]:

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

```
In [27]: %%sql
         select min(Date) from SPACEXTBL where "Landing _Outcome" = "Success (ground pad)"

 * sqlite:///my_data1.db
Done.
```

Out[27]:

min(Date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

In [29]:

```
%%sql
select distinct Booster_Version from SPACEXTBL
where "Landing _Outcome" = "Success (drone ship)"
and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_< 6000
```

```
 * sqlite:///my_data1.db
Done.
```

Out[29]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

In [32]:

```
%%sql
select Mission_Outcome, count(Mission_Outcome) from SPACEXTBL
group by Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Out[32]:

Mission_Outcome	count(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

In [31]:

```
%%sql
select distinct Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ =
(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

Out[31]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

In [35]:

```
%%sql
select substr(Date, 4, 2) as month,Booster_Version,Launch_Site,count("Landing _Outcome")
    from SPACEXTBL where substr(Date,7,4)='2015' and "Landing _Outcome" = "Failure (drone ship)"
    group by 1,2,3
```

```
 * sqlite:///my_data1.db
Done.
```

Out[35]:

month	Booster_Version	Launch_Site	count("Landing _Outcome")
01	F9 v1.1 B1012	CCAFS LC-40	1
04	F9 v1.1 B1015	CCAFS LC-40	1

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

In [56]:

```
%%sql
select "Landing _Outcome" , count("Landing _Outcome")  from SPACEXTBL
where Date between '04-06-2010' and '20-03-2017'
and "Landing _Outcome" like '%Success%'
group by "Landing _Outcome"
order by count("Landing _Outcome") desc
```

```
 * sqlite:///my_data1.db
Done.
```

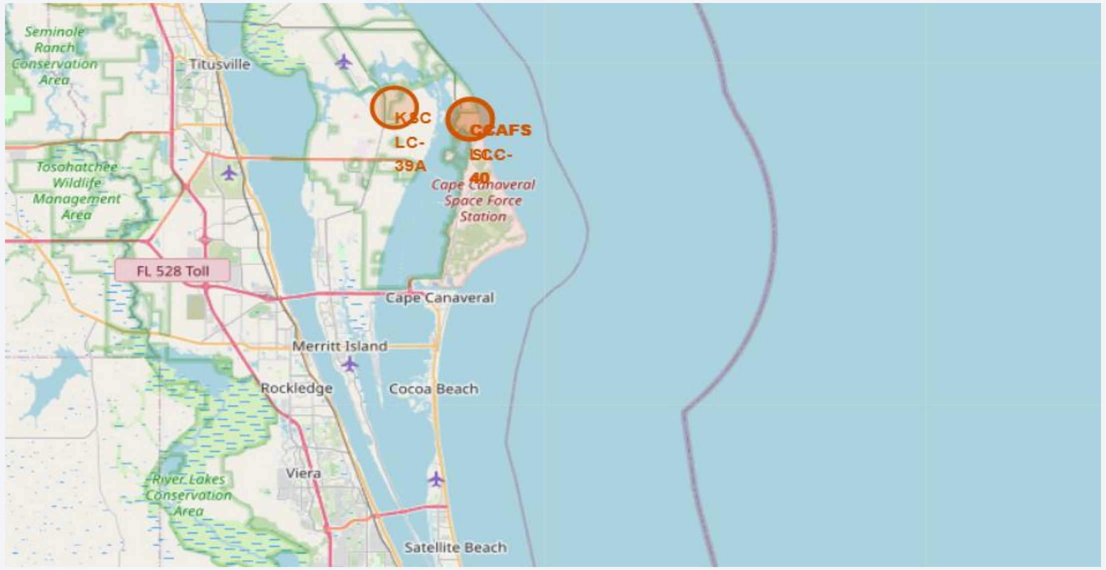
Out[56]:

Landing _Outcome	count("Landing _Outcome")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

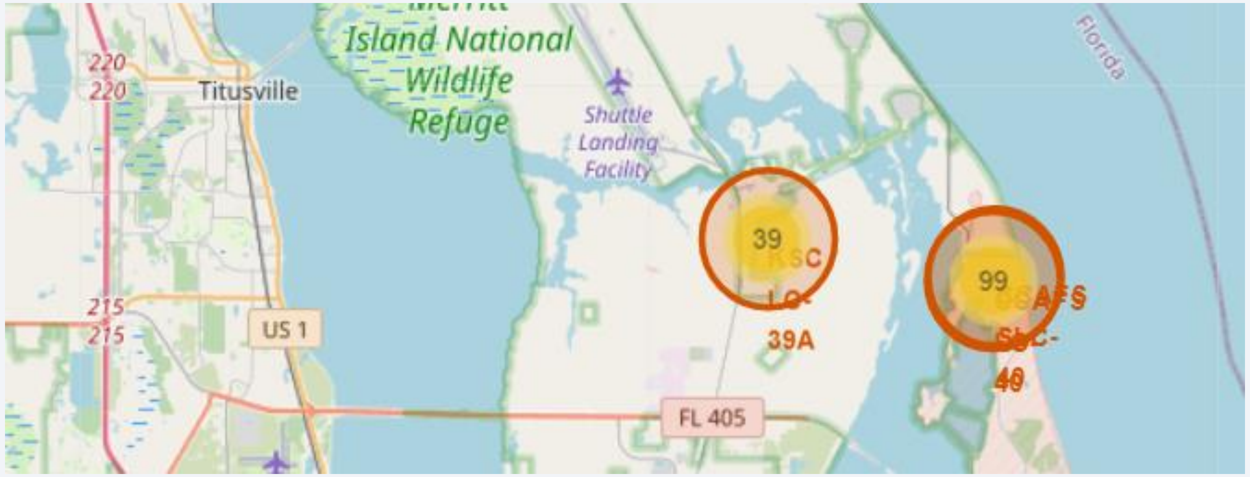
Launch Sites Proximities Analysis

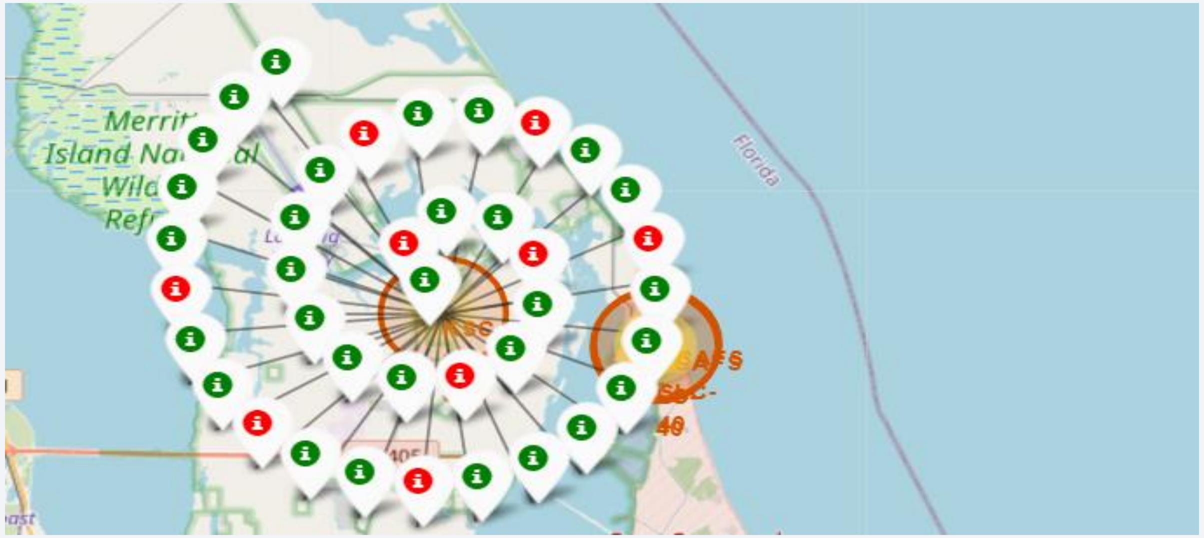
Launch sites on a Folium Map



- All the launch sites are in very close proximity to the coast

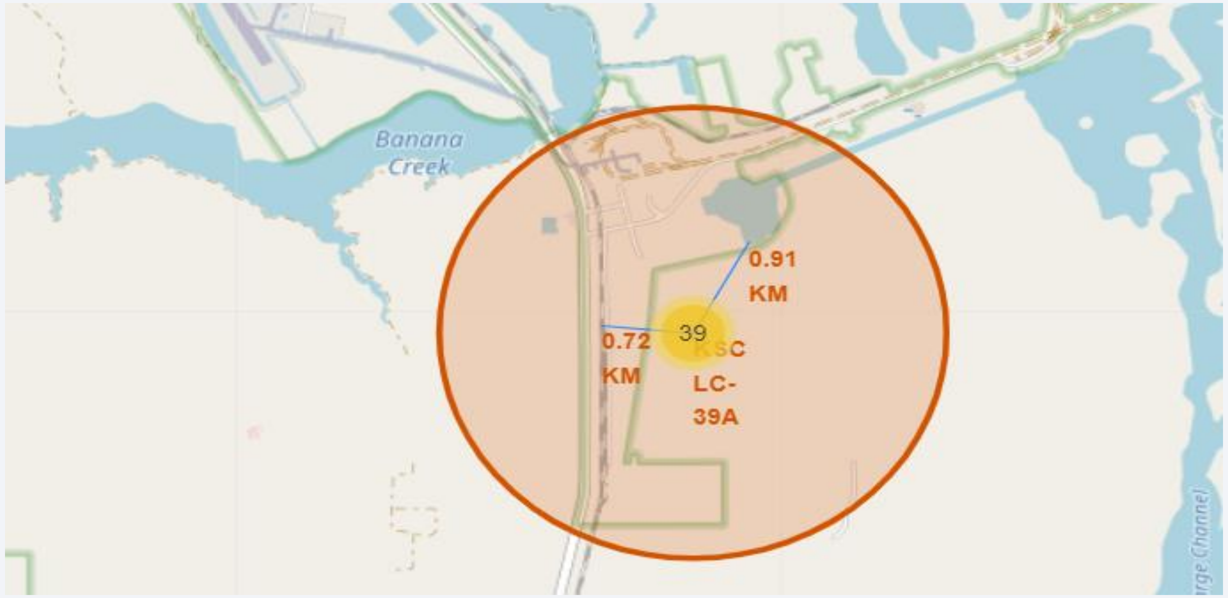
Success and Failed launches at each Launch site





- KSC LC-39A had the most successful launches among all the other sites
- Launches from the site CCAFS SLC 40 are significantly higher than the launches from other sites.

Distance between a launch site to its proximities



- The Launch sites are away from the cities
- The Launch sites were in proximity to railways compared to highways and coast

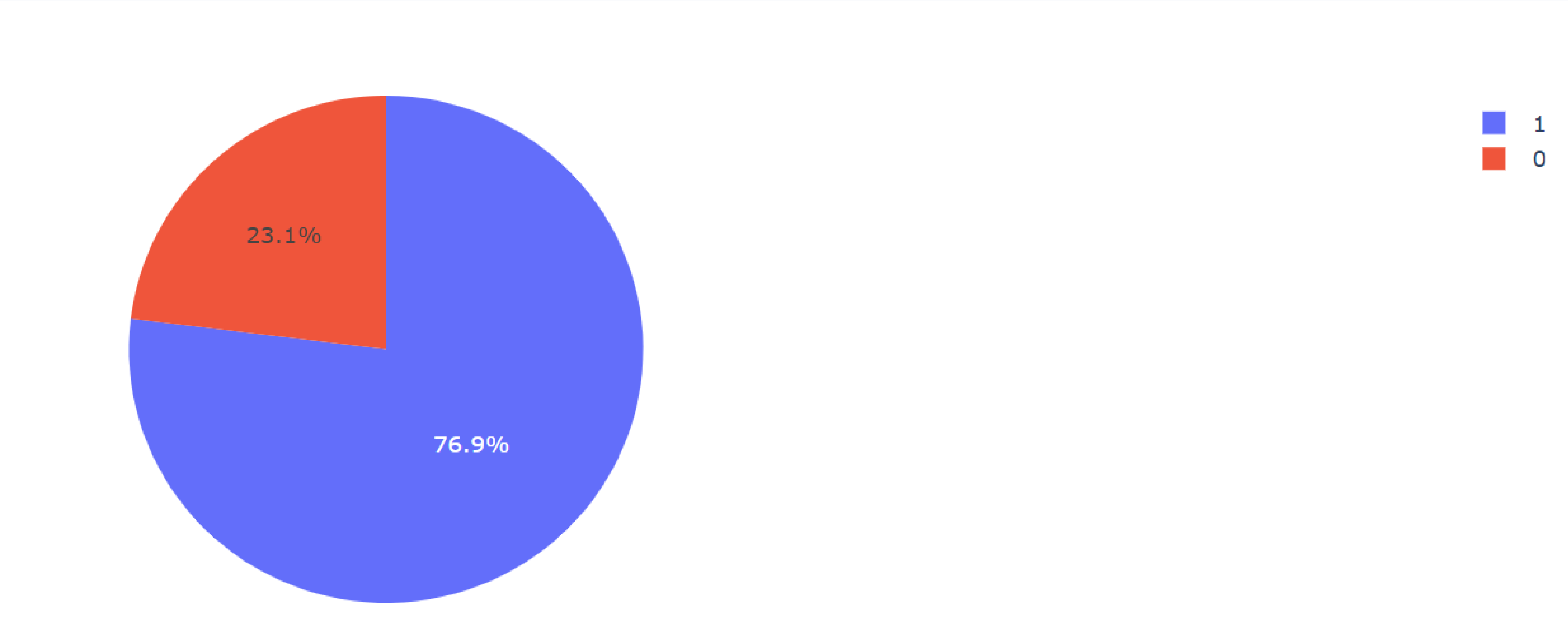
Section 4

Build a Dashboard with Plotly Dash

Percentage of success launches at each site

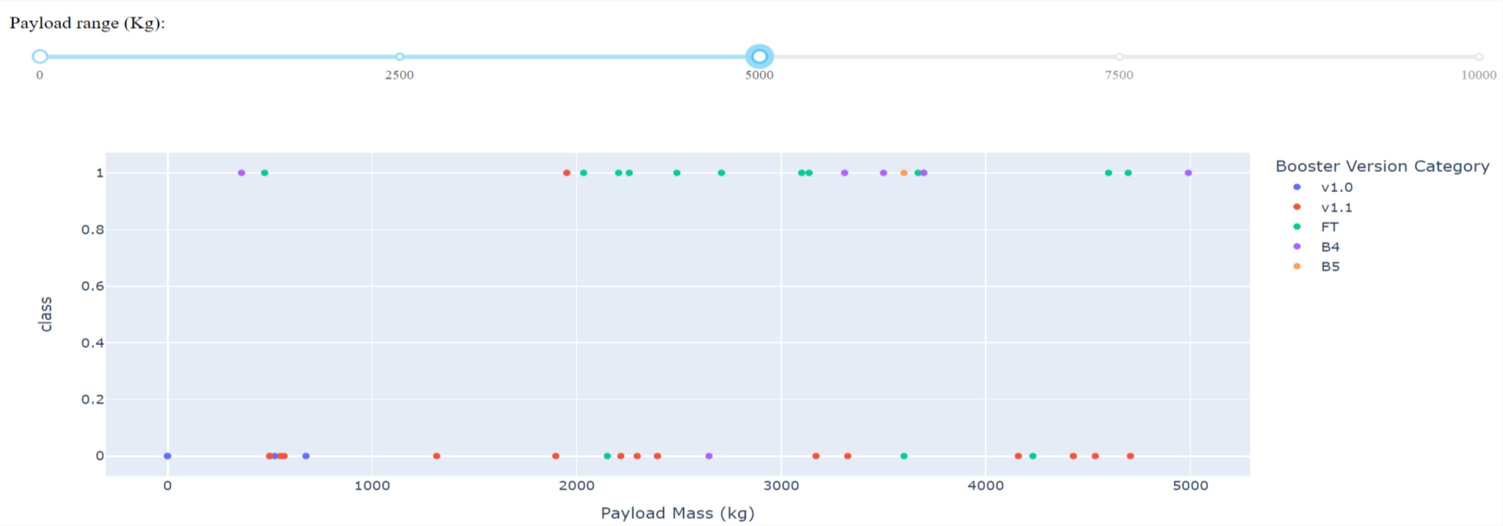
- KSC LC-39A had the most successful launches among all the other sites
- CCAFS LC-40 had the second most successful launches.

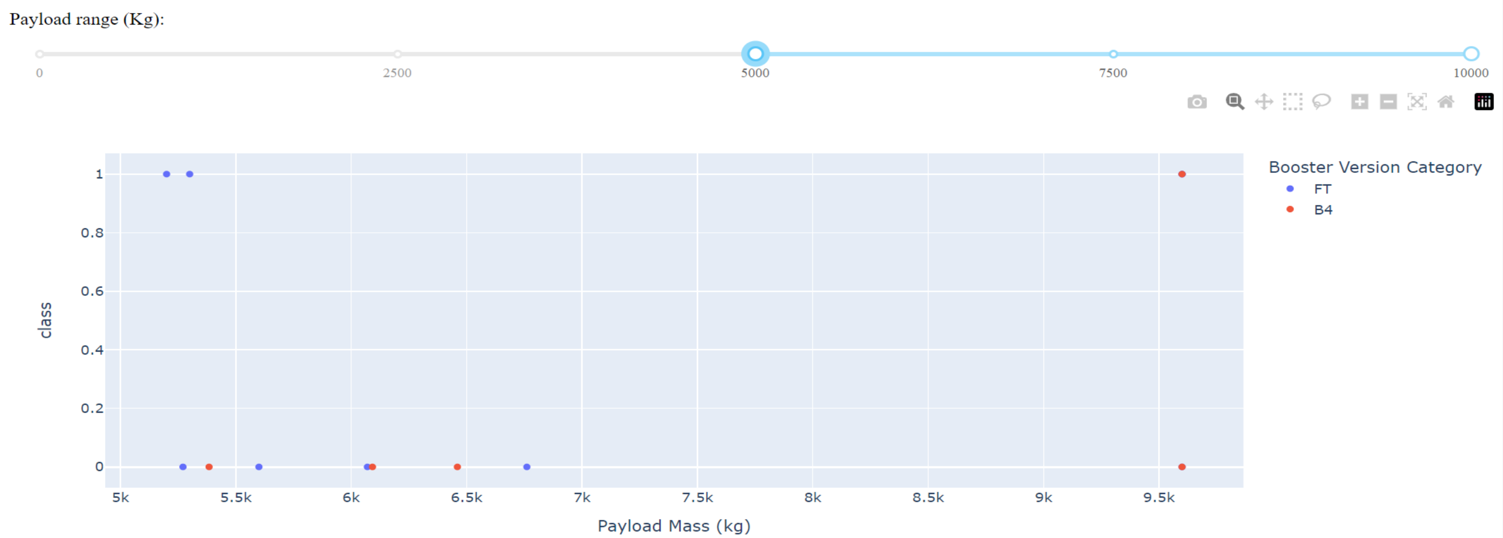
Launch site with highest success rate



- KSC LC-39A achieved 76.9% of success rate while getting a failure rate of 23.1%

Payload vs. Launch Outcome



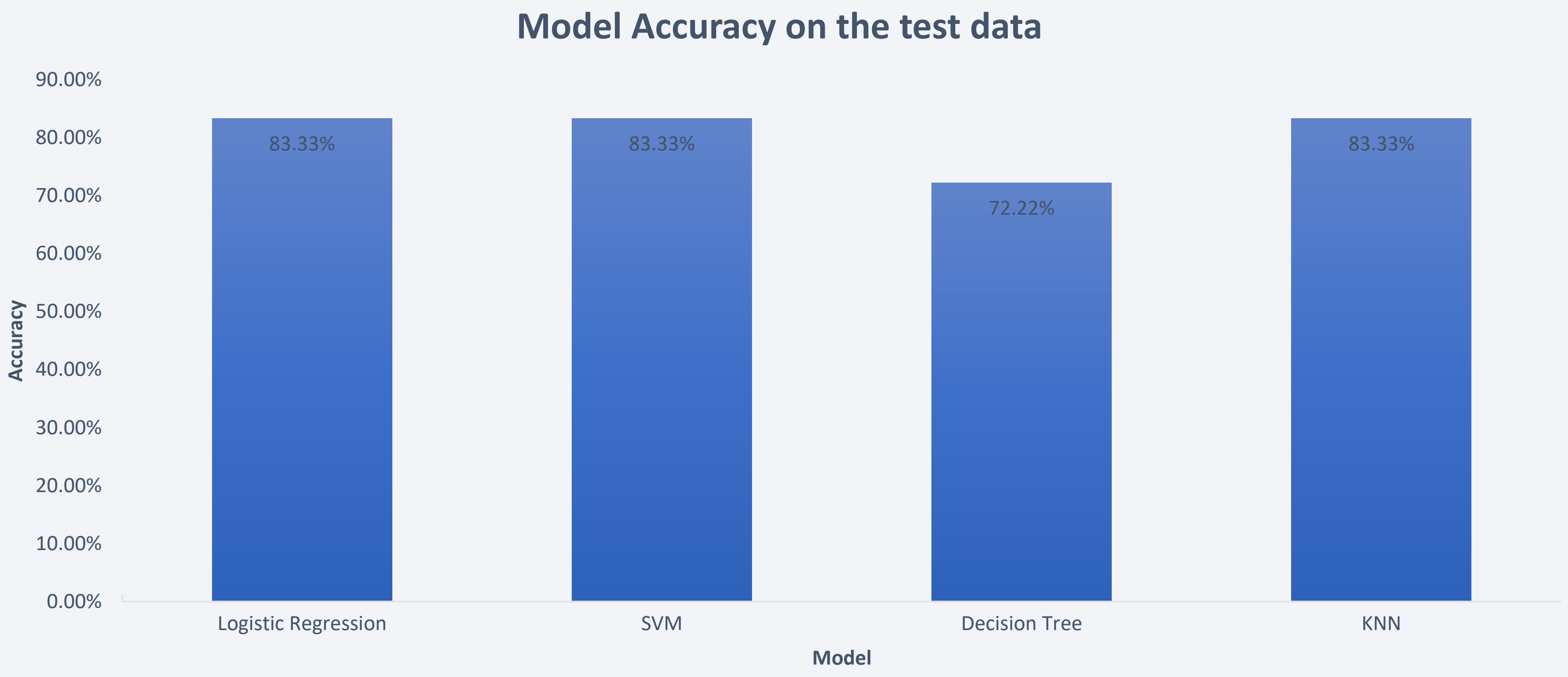


- Success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

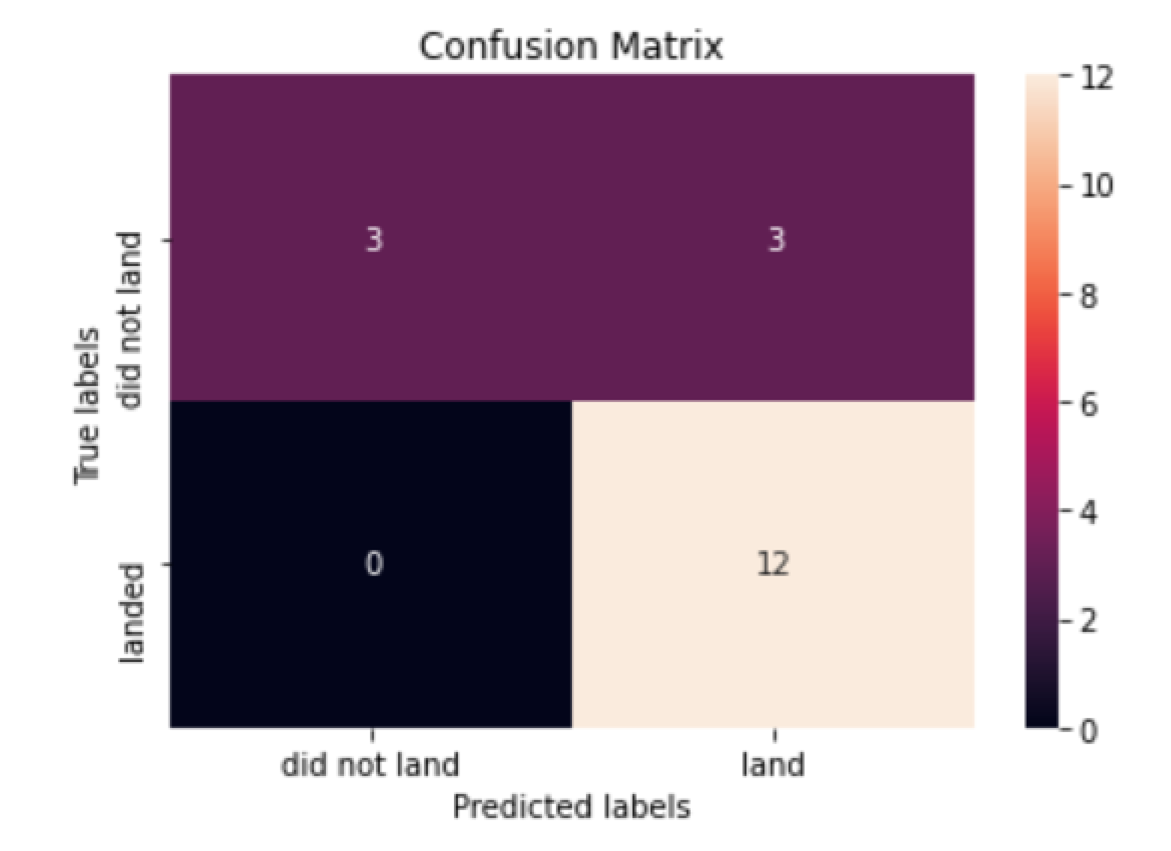
Predictive Analysis (Classification)

Classification Accuracy



- The LR, KNN and SVM achieved the highest accuracy of 83.33%

Confusion Matrix



- The Major problem here is False positives where the SpaceX launch did not land successfully but it was predicted positive (landed successfully)
- TP : 12 ; TN : 3 , FP : 3 , FN : 0
- Sensitivity : TP/TP+FN = 1.00
- Sensitivity : TN/FP+TN = 0.50
- Precision : TP/ TP+FP = 0.80
- Accuracy : (TP+TN)/Total : 15/18 = 0.83
- F1 score : 2TP/(2TP+FP+FN) = 0.89

Conclusions

- ES-L1, GEO, HEO and SSO orbits have the highest success rates
- The SpaceX launch success rate has been almost consistently increasing since 2013 except for a drop in 2018
- KSC LC-39A had the most successful launches among all the other sites
- Success rates for low weighted payloads is higher than the heavy weighted payloads
- The LR, KNN and SVM achieved the highest accuracy of 83.33%
- The Launch sites were in proximity to railways compared to highways and coast
- Launches from the site CCAFS SLC 40 are significantly higher than the launches from other sites.

Thank you!